Attorney Docket No. 1538.1020

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Shigeo ORII

Application No.: Group Art Unit:

Filed: November 30, 2001 Examiner:

For: PARALLEL EFFICIENCY CALCULATING METHOD AND APPARATUS

SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN APPLICATION IN ACCORDANCE WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2001-241121

Filed: August 8, 2001

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing date(s) as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: November 30, 2001 By: _____

James D. Halsey, Jr.
Registration No. 22,729

700 11th Street, N.W., Ste. 500
Washington, D.C. 20001
(202) 434-1500

日　本　国　特　許　庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日
Date of Application: ２００１年　８月　８日

出願番号
Application Number: 特願２００１－２４１１２１

出願人
Applicant(s): 富士通株式会社

２００１年　９月２１日

特許庁長官
Commissioner,
Japan Patent Office

及川耕造

出証番号　出証特２００１－３０８７６５５

【書類名】　特許願
【整理番号】　0151378
【提出日】　平成13年 8月 8日
【あて先】　特許庁長官殿
【国際特許分類】　G06F 09/28
【発明の名称】　並列効率計算方法及び装置
【請求項の数】　10
【発明者】
　【住所又は居所】　神奈川県川崎市中原区上小田中４丁目１番１号　富士通株式会社内
　【氏名】　折居　茂夫
【特許出願人】
　【識別番号】　000005223
　【氏名又は名称】　富士通株式会社
【代理人】
　【識別番号】　100103528
　【弁理士】
　【氏名又は名称】　原田　一男
　【電話番号】　045-290-2761
【手数料の表示】
　【予納台帳番号】　076762
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
　【包括委任状番号】　9909129
【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　並列効率計算方法及び装置

【特許請求の範囲】

【請求項１】

並列計算機システムの並列効率を計算する方法であって、

並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、前記並列処理プログラム実行時における並列処理部分の処理時間に関する情報と、並列処理のオーバーヘッドにより生ずる処理時間に関する情報とを取得し、記憶装置に格納するステップと、

取得された前記逐次処理部分の処理時間に関する情報と前記並列処理部分の処理時間に関する情報と前記並列処理のオーバーヘッドにより生ずる処理時間に関する情報とを用いて、並列化率と、逐次計算時間比と、並列オーバーヘッド比とを計算し、記憶装置に格納する指標計算ステップと、

計算された前記並列化率と前記逐次計算時間比と前記並列オーバーヘッド比とを用いて並列効率を計算し、記憶装置に格納する並列効率計算ステップと、

を含む並列効率計算方法。

【請求項２】

前記逐次処理部分の処理時間に関する情報が、前記並列処理プログラム実行期間内の所定周期毎の実行状況確認において逐次処理が実施されていると判断された回数であり、

前記並列処理部分の処理時間に関する情報が、前記並列処理プログラム実行期間内の所定周期毎の実行状況確認において並列処理が実施されていると判断された回数であり、

前記並列処理のオーバーヘッドにより生ずる処理時間に関する情報が、前記並列処理プログラム実行期間内の所定周期毎の実行状況確認において前記並列処理のオーバーヘッドにより生ずる処理が実施されていると判断された回数である

ことを特徴とする請求項１記載の並列効率計算方法。

【請求項３】

前記指標計算ステップが、

前記並列処理部分の処理時間に関する情報をプロセッサ数倍し、前記並列処理プログラム実行時における並列処理部分の逐次処理時における処理時間に関する情報として記憶装置に格納するステップと、

（前記並列処理部分の逐次処理時における処理時間に関する情報）／（前記逐次処理部分の処理時間に関する情報＋前記並列処理部分の逐次処理時における処理時間に関する情報）を計算し、前記並列化率として記憶装置に格納するステップと、

を含む請求項１記載の並列効率計算方法。

【請求項４】

前記指標計算ステップが、

（前記逐次処理部分の処理時間に関する情報）／（前記並列処理プログラムの全処理時間に関する情報）を計算し、前記逐次計算時間比として記憶装置に格納するステップ、

を含む請求項１記載の並列効率計算方法。

【請求項５】

前記指標計算ステップが、

（前記並列処理のオーバーヘッドにより生ずる処理時間に関する情報）／（前記並列処理プログラムの全処理時間に関する情報）を計算し、前記並列オーバーヘッド比として記憶装置に格納するステップ、

を含む請求項１記載の並列効率計算方法。

【請求項６】

前記並列効率計算ステップが、

１／（前記並列化率）×（１－（前記逐次計算時間比）－（前記並列オーバーヘッド比））を計算し、前記並列効率として記憶装置に格納するステップ、

であることを特徴とする請求項１記載の並列効率計算方法。

【請求項７】

並列計算機システムの並列効率を計算する方法であって、

並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、前記並列処理プログラム実行時における並列処理部分の処理時間に関する情報と

前記並列処理プログラムの全処理時間に関する情報とを取得し、記憶装置に格納するステップと、

取得された前記並列処理部分の処理時間に関する情報をプロセッサ数倍し、前記並列処理プログラム実行時における並列処理部分の逐次処理時における処理時間に関する情報として記憶装置に格納するステップと、

取得された前記逐次処理部分の処理時間に関する情報と前記並列処理部分の処理時間に関する情報とを少なくとも用いて、並列化率と、逐次計算時間比と、並列オーバーヘッド比とを計算し、記憶装置に格納する指標計算ステップと、

（（前記逐次処理部分の処理時間に関する情報）＋（前記並列処理部分の逐次処理時における処理時間に関する情報））／（（前記並列処理プログラムの全処理時間に関する情報）×（前記プロセッサ数））を計算し、並列効率として記憶装置に格納するステップと、

を含む並列効率計算方法。

【請求項８】

並列計算機システムの並列効率を計算する方法であって、

並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、前記並列処理プログラム実行時における並列処理部分の処理時間に関する情報と前記並列処理プログラムの全処理時間に関する情報とを取得し、記憶装置に格納するステップと、

取得された前記逐次処理部分の処理時間に関する情報と前記並列処理部分の処理時間に関する情報とを用いて並列化率を計算し、記憶装置に格納するステップと、

前記並列化率の逆数と前記並列処理プログラムの全処理時間に関する情報の逆数と前記並列処理部分の処理時間に関する情報との積を計算し、並列効率として記憶装置に格納するステップと、

を含む並列効率計算方法。

【請求項９】

並列計算機システムの並列効率を計算するプログラムであって、

並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、

前記並列処理プログラム実行時における並列処理部分の処理時間に関する情報と、並列処理のオーバーヘッドにより生ずる処理時間に関する情報とを取得し、記憶装置に格納するステップと、

取得された前記逐次処理部分の処理時間に関する情報と前記並列処理部分の処理時間に関する情報と前記並列処理のオーバーヘッドにより生ずる処理時間に関する情報とを用いて、並列化率と、逐次計算時間比と、並列オーバーヘッド比とを計算し、記憶装置に格納する指標計算ステップと、

計算された前記並列化率と前記逐次計算時間比と前記並列オーバーヘッド比とを用いて並列効率を計算し、記憶装置に格納する並列効率計算ステップと、

をコンピュータに実行させるためのプログラム。

【請求項１０】

並列計算機システムの並列効率を計算するプログラムであって、

並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、前記並列処理プログラム実行時における並列処理部分の処理時間に関する情報と前記並列処理プログラムの全処理時間に関する情報とを取得し、記憶装置に格納するステップと、

取得された前記並列処理部分の処理時間に関する情報をプロセッサ数倍し、前記並列処理プログラム実行時における並列処理部分の逐次処理時における処理時間に関する情報として記憶装置に格納するステップと、

取得された前記逐次処理部分の処理時間に関する情報と前記並列処理部分の処理時間に関する情報とを少なくとも用いて、並列化率と、逐次計算時間比と、並列オーバーヘッド比とを計算し、記憶装置に格納する指標計算ステップと、

（（前記逐次処理部分の処理時間に関する情報）＋（前記並列処理部分の逐次処理時における処理時間に関する情報））／（（前記並列処理プログラムの全処理時間に関する情報）×（前記プロセッサ数））を計算し、並列効率として記憶装置に格納するステップと、

をコンピュータに実行させるためのプログラム。

【発明の詳細な説明】

【０００１】

【発明が属する技術分野】

本発明は、並列計算機システムの性能評価技術に関する。

【０００２】

【従来の技術】

従来の並列計算機システムの性能評価は、以下に示す並列効率$E_{para}(p,n)$を求めることにより行われていた。

【数１】

$$E_{para}(p,n) \equiv \frac{\tau(1,n)}{\tau(p,n)\cdot p} \quad (1)$$

ここでｐはプロセッサ数、ｎは問題の大きさである。式（１）により並列効率$E_{para}(p,n)$を求めるためには、逐次処理を行った場合の処理時間である逐次処理時間$\tau(1,n)$及び並列処理を行った場合の処理時間である並列処理時間$\tau(p,n)$を個別に測定することが必要となる。但し、並列処理時間$\tau(p,n)$が長時間になると、それよりも長くなる逐次処理時間$\tau(1,n)$を測定することが困難である場合も生じる。

【０００３】

また、この並列効率$E_{para}(p,n)$が低い場合、すなわち並列処理の性能が悪い場合には、性能改善を阻害する要因を特定することがその改善のために必要となる。このため更に１回以上の測定を行い、並列処理時間に対する性能改善を阻止する要因の割合を調べてその要因を特定しなければならなかった。

【０００４】

【発明が解決しようとする課題】

従来の評価方法では、性能改善を阻害する要因と並列効率との定量関係が明確ではなかったため、どの阻害要因がどのくらい並列効率に効いているかを判断することが難しかった。

【０００５】

よって本発明の目的は、並列効率の値と性能改善を阻害する要因を定量的に結び付けることにより性能を阻害している原因を明確にするための技術を提供することである。

【０００６】

また、本発明の目的は、並列効率を１回の測定により計算できるようにするための技術を提供することである。

【０００７】

さらに、本発明の目的は、より精度よく並列効率を計算できるようにするための技術を提供することである。

【０００８】

また、並列計算機システムの性能評価を容易にし且つ評価に要する時間を短縮できるようにするための技術を提供することも本発明の目的である。

【０００９】

【課題を解決するための手段】

本発明の第１の態様に係る、並列計算機システムの並列効率を計算する方法は、並列処理プログラム実行時における逐次処理部分の処理時間（例えば実施の形態における$\alpha(p,n)$）に関する情報（例えば処理時間又は事象の確認回数）と、並列処理プログラム実行時における並列処理部分の処理時間（例えば実施の形態における$\beta(p,n)/p$）に関する情報（例えば処理時間又は事象の確認回数）と、並列処理のオーバーヘッドにより生ずる処理時間（例えば実施の形態における$\sigma(p,n)$）に関する情報（例えば処理時間又は事象の確認回数）とを取得し、記憶装置に格納するステップと、取得された逐次処理部分の処理時間に関する情報と並列処理部分の処理時間に関する情報と並列処理のオーバーヘッドにより生ずる処理時間に関する情報とを用いて、並列化率（例えば実施の形態における$R_{para}(p,n)$）と、逐次計算時間比（例えば実施の形態における$R_{\alpha}(p,n)$）と、並列オーバーヘッド比（例えば実施の形態における$R_{\sigma}(p,n)$）とを計算し、記憶装置に格納する指標計算ステップと、計算された並列化率と逐次計算時間比と並列オーバーヘッド比とを用いて並列効率（E_{para}）を計算し、記憶装置に格納する並列効率計算ステップとを含む。最後に、計算結果を表示装置等に出力する場合もある。

【００１０】

並列効率が、並列化率、逐次計算時間比及び並列オーバーヘッド比という性能

改善を阻害する要因の指標により表されるため、これらの指標の値と並列効率の値とを用いて性能改善要因を特定し、性能改善のための対策を決定することができるようになる。

【００１１】

また、並列処理プログラム実行時における逐次処理部分の処理時間と、並列処理プログラム実行時における並列処理部分の処理時間と、並列処理のオーバーヘッドにより生ずる処理時間とは、１回の処理実行に対する時間等の計測にて取得できるため、当該並列計算機の性能評価を容易にし且つ評価に要する時間を短縮することができるようになる。

【００１２】

なお、上で述べた並列効率計算ステップを、１／（並列化率）×（１－（逐次計算時間比）－（並列オーバーヘッド比））を計算し、並列効率として記憶装置に格納するステップとすることも可能である。

【００１３】

本発明の第２の態様に係る並列効率計算方法は、並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、並列処理プログラム実行時における並列処理部分の処理時間に関する情報と並列処理プログラムの全処理時間に関する情報とを取得し、記憶装置に格納するステップと、取得された並列処理部分の処理時間に関する情報をプロセッサ数倍し、並列処理プログラム実行時における並列処理部分の逐次処理時における処理時間（例えば実施の形態における$\beta(p,n)$）に関する情報として記憶装置に格納するステップと、取得された逐次処理部分の処理時間に関する情報と並列処理部分の処理時間に関する情報とを少なくとも用いて、並列化率と、逐次計算時間比と、並列オーバーヘッド比とを計算し、記憶装置に格納する指標計算ステップと、（（逐次処理部分の処理時間に関する情報）＋（並列処理部分の逐次処理時における処理時間に関する情報））／（（並列処理プログラムの全処理時間に関する情報）×（プロセッサ数））を計算し、並列効率として記憶装置に格納するステップとを含む。最後に計算結果を表示装置等に表示する場合もある。

【００１４】

このようにすれば、１回の処理実行に対する時間等の計測にて並列効率を計算できるため、当該並列計算機の性能評価を容易にし且つ評価に要する時間を短縮することができるようになる。また、計算精度を向上させることもできる。

【００１５】

本発明の第３の態様に係る並列効率計算方法は、並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、並列処理プログラム実行時における並列処理部分の処理時間に関する情報と並列処理プログラムの全処理時間に関する情報とを取得し、記憶装置に格納するステップと、取得された逐次処理部分の処理時間に関する情報と並列処理部分の処理時間に関する情報とを用いて並列化率を計算し、記憶装置に格納するステップと、並列化率の逆数と並列処理プログラムの全処理時間に関する情報の逆数と並列処理部分の処理時間に関する情報との積を計算し、並列効率として記憶装置に格納するステップとを含む。最後に計算結果を表示装置に表示する場合もある。

【００１６】

このような計算方法にても、１回の処理実行に対する時間等の計測にて並列効率を計算できるため、当該並列計算機の性能評価を容易にし且つ評価に要する時間を短縮することができるようになる。

【００１７】

なお、上述の並列効率計算方法はコンピュータ・ハードウエアに専用のプログラムをインストールすることによっても実現可能である。この場合、このプログラムは、例えばフレキシブルディスク、ＣＤ－ＲＯＭ、光磁気ディスク、半導体メモリ、ハードディスク等の記憶媒体又は記憶装置に格納される。また、ネットワークなどを介して配布される場合もある。尚、中間的な処理結果はメモリに一時保管される。

【００１８】

【発明の実施の形態】

［本発明の原理］

並列計算機システムの各プロセッサにおいて実行すべき処理量に偏りが無い場合（ロードインバランスが無い場合）の並列計算機システムの処理時間の関係は

、以下のように表される。

$\tau(p,n) \equiv \alpha(p,n) + \beta(p,n) / p + \sigma(p,n)$　　（２）

ここでｐはプロセッサ数である。ｎは問題の規模で、例えば粒子シミュレーションでは粒子数或いは構造解析では要素数を意味し、その値は現在百万のオーダーにも及ぶものがあり、更に時代と共に大きくなりつつある。また、$\alpha(p,n)$は並列実行時における逐次処理部分の処理時間であり、$\beta(p,n)/p$は並列実行時における並列処理部分の処理時間であり、$\sigma(p,n)$は並列処理のオーバーヘッドにより生ずる処理時間である。すなわち、並列処理時間（並列処理時の全処理時間とも呼ぶ）$\tau(p,n)$は、逐次処理部分の処理時間と、並列処理部分の処理時間と、並列処理のオーバーヘッドにより生ずる処理時間との和である。

【００１９】

プロセッサが１つ（ｐ＝１）の場合には、並列処理部分の処理時間はプロセッサ数ｐ倍になり、並列処理のオーバーヘッドが無くなるため、以下のように表される。

$\tau(1,n) = \alpha(p,n) + \beta(p,n)$　　（３）

すなわち、逐次処理時間（逐次処理時の全処理時間とも呼ぶ）$\tau(1,n)$は、逐次処理部分の処理時間と並列処理部分の処理時間をプロセッサ数倍したものとの和となる。

【００２０】

また、並列性能を決定する要因として、

並列化率：$R_{para}(p,n) \equiv \beta(p,n) / [\alpha(p,n) + \beta(p,n)]$　　（４）

逐次計算時間比：$R_{\alpha}(p,n) \equiv \alpha(p,n) / \tau(p,n)$　　（５）

並列オーバーヘッド比：$R_{\sigma}(p,n) \equiv \sigma(p,n) / \tau(p,n)$　　（６）

という３つの指標を導入する。

【００２１】

並列化率$R_{para}(p,n)$は、並列処理部分を逐次処理した場合の処理時間（並列処理部分の処理時間をプロセッサ数倍したもの）を逐次処理時間と並列処理部分を逐次処理した場合の処理時間との和で除したものである。この並列化率$R_{para}(p,n)$は、大きな値を有すれば有するほど（１に近ければ近いほど）並列処理を

行っている割合が高いことを示している。逐次計算時間比$R_\alpha(p,n)$は、逐次処理部分の処理時間を並列処理時間で除したものである。この逐次計算時間比$R_\alpha(p,n)$は、大きな値であるほど並列処理できない逐次処理部分の処理時間の割合が高いことを示している。並列オーバーヘッド比$R_\sigma(p,n)$は、並列処理のオーバーヘッドにより生ずる処理時間を並列処理時間で除したものである。この並列オーバーヘッド比$R_\sigma(p,n)$は、大きな値であるほど並列処理のオーバーヘッドにより生ずる処理時間の割合が高いことを示している。

【００２２】

式（２）乃至（６）を従来技術の欄で示した式（１）に代入すると、式（１）で表される並列効率$E_{para}(p,n)$は以下のように変形される。

【数２】

$$E_{para}(p,n)=\frac{1}{R_{para}(p,n)}\cdot(1-R_\alpha(p,n)-R_\sigma(p,n)) \quad (7)$$

式（７）の右辺を見れば、並列効率$E_{para}(p,n)$は、並列化率$R_{para}(p,n)$の逆数に１から逐次計算時間比$R_\alpha(p,n)$と並列オーバーヘッド比$R_\sigma(p,n)$とを減じた値を乗じた値となる。このように、並列化率$R_{para}(p,n)$、逐次計算時間比$R_\alpha(p,n)$及び並列オーバーヘッド比$R_\sigma(p,n)$という３つの性能改善を阻害する要因の指標のみで並列効率$E_{para}(p,n)$は定量的に表されることができるようになった。

【００２３】

なお、式（７）は以下のような式にも変形することができる。

【数３】

$$E_{para}(p,n)=\frac{1}{R_{para}(p,n)}\cdot\frac{1}{\tau(p,n)}\cdot\frac{\beta(p,n)}{p} \quad (8)$$

【数４】

$$E_{para}(p,n) \equiv \frac{\alpha(p,n)+\beta(p,n)}{\tau(p,n)\cdot p} \quad (9)$$

【００２４】

式（８）の右辺は、並列効率$E_{para}(p,n)$が、並列化率$R_{para}(p,n)$の逆数と、並列処理時間$\tau(p,n)$の逆数と、並列処理部分の処理時間$\beta(p,n)$／ｐとの積により計算されることを示している。

【００２５】

また式（９）の右辺は、並列効率$E_{para}(p,n)$が、逐次処理部分の処理時間$\alpha(p,n)$と並列処理部分の逐次処理時における処理時間$\beta(p,n)$との和を並列処理時間$\tau(p,n)$とプロセッサ数ｐにより除することにより計算されることを示している。なお、計算精度の観点からすると、式（９）が最も好ましい事がわかっている。

【００２６】

［実施の形態の説明］

以上のような発明の原理を実施するための一実施の形態の説明を以下に行う。図１は、並列計算機システムの並列効率を計算するための並列効率計算装置であるコンピュータ１の機能ブロック図を示している。なお、コンピュータ１自身も並列計算機システムであってもよいが、コンピュータ１はプロセッサが１つのコンピュータであってもよい。また、並列計算機システムは、分散メモリ型のものであってもよいし、メモリを共有するＳＭＰ（Symmetric MultiProcessor）型のものであってもよい。

【００２７】

コンピュータ１は、データ取得部１１と、指標データ計算部１３と、並列効率計算部１５と、出力部１７と、表示装置や印刷装置等である出力装置１９とを含む。データ取得部１１は、逐次処理部分の処理時間$\alpha(p,n)$と、並列処理部分の処理時間$\beta(p,n)$／ｐと、並列処理のオーバーヘッドにより生ずる処理時間$\sigma(p,n)$と、並列処理時間$\tau(p,n)$と、プロセッサ数ｐとを取得する。なお、プロセッ

サ数ｐについては、ユーザからの入力や、評価対象の並列計算機システムから取得する。また、時間の情報を取得するとしていたが、逐次処理、並列処理及び並列処理のオーバーヘッドにより生ずる処理の各事象の出現頻度を用いること（以下、サンプリングの場合と呼ぶ）も可能である。例えば、並列処理プログラム実行期間内において所定周期毎に実行状況を確認して、各事象の出現回数をカウントすることにより出現頻度を取得する。このように出現頻度を用いることができるのは、式（７）、（８）及び（９）において出現する並列化率$R_{para}(p,n)$、逐次計算時間比$R_{\alpha}(p,n)$、並列オーバーヘッド比$R_{\sigma}(p,n)$、式（８）の$R_{para}(p,n)$の逆数以外の部分、及び式（９）自体が、時間比の形をしているためである。なお、時間の情報を用いる場合とサンプリングの場合とでは測定精度に差は生じえる。

【００２８】

時間の情報を取得する場合には、例えば各処理部分をタイマで挟み、実際に時間測定することにより取得することができる。すなわち、各処理部分の開始段階で時刻をメモリに記録し、またその終了段階で時刻をメモリに記録するようにして、並列処理プログラムを実行する。そして、開始段階における時刻と終了段階における時刻との差をデータ取得部１１で計算することにより、各処理部分の時間を取得できる。そして、データ取得部１１は、最終的に逐次処理部分の処理時間、並列処理部分の処理時間、並列処理のオーバーヘッドにより生ずる処理時間をそれぞれについて集計することにより、逐次処理部分の処理時間$\alpha(p,n)$と、並列処理部分の処理時間$\beta(p,n)$／ｐと、並列処理のオーバーヘッドにより生ずる処理時間$\sigma(p,n)$を得ることができる。また、逐次処理部分の処理時間$\alpha(p,n)$と、並列処理部分の処理時間$\beta(p,n)$／ｐと、並列処理のオーバーヘッドにより生ずる処理時間$\sigma(p,n)$との和を計算することにより、データ取得部１１は、並列処理時間$\tau(p,n)$を取得することができる。なお、並列処理時間$\tau(p,n)$については、処理開始時刻と処理終了時刻とを記録し、その差を計算することによっても得ることができる。

【００２９】

サンプリングの場合、一定時間間隔毎に実行中の並列処理プログラムの事象（

逐次処理、並列処理、並列処理のオーバーヘッドにより生ずる処理）を識別し、事象毎にカウントを行う。この事象の識別とカウントは、並列計算機システムのハードウエア又はプログラムにて行われる。例えば、（１）逐次処理の事象については、（ａ）並列処理プログラムにおいて逐次処理部分を実行する場合にそれを表すフラグを立てるようにし、当該並列処理プログラム実行中の実行状況確認時にこのフラグが立っていれば、逐次処理についてのカウントを１インクリメントする。（ｂ）また、全体のカウント値から並列処理及び並列処理のオーバーヘッドにより生ずる処理についてのカウント値を差し引くことにより逐次処理についてのカウント値を得ることも可能である。

【００３０】

（２）並列処理の事象については、（ａ）並列処理プログラムにおいて並列処理部分を実行する場合にそれを表すフラグを立てるようにプログラミングし、当該並列処理プログラム実行中の実行状況確認時にこのフラグが立っていれば、並列処理についてのカウントを１インクリメントする。（ｂ）また、並列化用コンパイラ・ディレクティブを認識したコンパイラやツール等により並列処理実行時に並列処理についてのフラグを立てるようコンパイルし、並列処理プログラム実行中の実行状況確認時にこのフラグが立っていれば、並列処理についてのカウントを１インクリメントするような構成でも良い。（ｃ）さらに、並列言語拡張を認識したコンパイラやツール等により並列処理実行時に並列処理についてのフラグを立てるようコンパイルし、当該並列処理プログラム実行中の実行状況確認時にこのフラグが立っていれば、並列処理についてのカウントを１インクリメントするような構成でもよい。（ｄ）又、コンパイラがコンパイル時に自動的に並列処理を実行するように判断した場合には当該並列処理実行時に並列処理についてのフラグを立てるようコンパイルし、並列処理プログラム実行中の実行状況確認時にこのフラグが立っていれば、並列処理についてのカウントを１インクリメントするような構成でもよい。（ｅ）さらに、並列実行するモジュール名をリストにまとめ、並列処理プログラム実行中の実行状況確認時にモジュール名を識別して、並列処理中であれば並列処理についてのカウントを１インクリメントするような構成でもよい。（ｆ）並列実行するイベント名をリストにまとめ、並列処理

プログラム実行中の実行状況確認時にイベント名を識別して、並列処理中であれば並列処理についてのカウントを１インクリメントするようにしてもよい。

【００３１】

（３）並列処理のためのオーバーヘッドにより生ずる処理の事象については、（ａ）並列処理プログラムにおいて並列処理のためのオーバーヘッドにより生ずる処理部分を実行する場合にその範囲を示すフラグを立てるようにプログラミングし、当該並列処理プログラム実行中の実行状況確認時にこのフラグが立っていれば、並列処理のためのオーバーヘッドにより生ずる処理についてのカウントを１インクリメントする。（ｂ）また、通信用コンパイラ・ディレクティブを認識したコンパイラやツール等により通信処理実行時に並列処理のためのオーバーヘッドにより生ずる処理についてのフラグを立てるようコンパイルし、並列処理プログラム実行中の実行状況確認時にこのフラグが立っていれば、並列処理のオーバーヘッドのための処理についてのカウントを１インクリメントするような構成でもよい。（ｃ）さらに、並列言語拡張を認識したコンパイラやツール等により通信処理実行時等に並列処理のためのオーバーヘッドにより生ずる処理についてのフラグを立てるようコンパイルし、当該並列処理プログラム実行中の実行状況確認時にこのフラグが立っていれば、並列処理のためのオーバーヘッドにより生ずる処理についてのカウントを１インクリメントするような構成でもよい。（ｄ）また、通信ライブラリ名を識別しておき、この通信ライブラリが実行された場合には、並列処理のためのオーバーヘッドにより生ずる処理についてのカウントを１インクリメントするような構成でもよい。（ｅ）さらに、コンパイラがコンパイル時に自動的に並列処理のためのオーバーヘッドにより生ずる処理を識別し、並列処理のためのオーバーヘッドにより生ずる処理実行時にそのためのフラグを立てるようコンパイルし、並列処理プログラム実行中の実行状況確認時にこのフラグが立っていれば、並列処理のためのオーバーヘッドにより生ずる処理についてのカウントを１インクリメントするような構成でもよい。（ｆ）さらに、通信で使用するモジュール名をリストにまとめ、並列処理プログラム実行中の実行状況確認時にモジュール名を識別して、通信処理中であれば並列処理のためのオーバーヘッドにより生ずる処理についてのカウントを１インクリメントするよう

にしてもよい。（ｇ）通信のイベント名をリストにまとめ、並列処理プログラム実行中の実行状況確認時にイベント名を識別して、通信処理中であれば並列処理のためのオーバーヘッドにより生ずる処理についてのカウントを１インクリメントするような構成であってもよい。

【００３２】

どの場合であっても並列処理プログラム実行中に直接カウントするのではなく、フラグや実行モジュール、イベント等の履歴を残しておき、後から一定時間間隔毎に事象を識別し、カウントを行うようにしても良い。

【００３３】

指標データ計算部１３は、データ取得部１１で取得された逐次処理部分の処理時間$\alpha(p,n)$と、並列処理部分の処理時間$\beta(p,n)/p$と、並列処理のオーバーヘッドにより生ずる処理時間$\sigma(p,n)$と、並列処理時間$\tau(p,n)$と、プロセッサ数ｐとを用いて、並列化率$R_{para}(p,n)$と、逐次計算時間比$R_{\alpha}(p,n)$と、並列オーバーヘッド比$R_{\sigma}(p,n)$とを、式（４）、（５）及び（６）を用いて計算する。なお、並列処理部分を逐次処理した場合の処理時間$\beta(p,n)$は、並列処理部分の処理時間$\beta(p,n)/p$をプロセッサ数ｐ倍することにより得られ、この値も用られる。

【００３４】

並列効率計算部１５は、指標データ計算部１３により計算された並列化率$R_{para}(p,n)$と、逐次計算時間比$R_{\alpha}(p,n)$と、並列オーバーヘッド比$R_{\sigma}(p,n)$とを用いて、式（７）に従って並列効率$E_{para}(p,n)$を計算する。

【００３５】

ここでは出力部１７は、指標データ計算部１３により計算された並列化率$R_{para}(p,n)$と、逐次計算時間比$R_{\alpha}(p,n)$と、並列オーバーヘッド比$R_{\sigma}(p,n)$と、並列効率計算部１５により計算された並列効率$E_{para}(p,n)$を、表示装置や、印刷装置等の出力装置に出力する。

【００３６】

次に、図２を用いてコンピュータ１の処理フローの一例を示す。最初に、コンピュータ１のデータ取得部１１は、プロセッサ数ｐ、逐次処理部分の処理時間α

(p,n)、並列処理部分の処理時間β(p,n)／ｐ、並列処理のオーバーヘッドにより生ずる処理時間σ(p,n)及び並列処理時間τ(p,n)を取得し、メインメモリ等の記憶装置に格納する（ステップＳ１）。また、例えばデータ取得部１１は、並列処理部分の処理時間β(p,n)／ｐにプロセッサ数ｐを掛けて、並列処理部分を逐次処理した場合の処理時間β(p,n)を計算し、記憶装置に格納する（ステップＳ３）。

【００３７】

次に、指標データ計算部１３は、並列化率$R_{para}(p,n)$を式（４）に従って計算し、計算結果を記憶装置に格納する（ステップＳ５）。また、指標データ計算部１３は、逐次計算時間比$R_{\alpha}(p,n)$を式（５）に従って計算し、計算結果を記憶装置に格納する（ステップＳ７）。さらに、指標データ計算部１３は、並列オーバーヘッド比$R_{\sigma}(p,n)$を式（６）に従って計算し、計算結果を記憶装置に格納する（ステップＳ９）。ステップＳ５乃至ステップＳ９の実行順番は任意である。

【００３８】

そして、並列効率計算部１５は、ステップＳ５乃至ステップＳ９において計算された並列化率$R_{para}(p,n)$、逐次計算時間比$R_{\alpha}(p,n)$及び並列オーバーヘッド比$R_{\sigma}(p,n)$を用いて式（７）に従って並列効率$E_{para}(p,n)$を計算し、記憶装置に格納する（ステップＳ１１）。出力部１７は、ステップＳ５乃至ステップＳ１１で計算した、並列化率$R_{para}(p,n)$、逐次計算時間比$R_{\alpha}(p,n)$、並列オーバーヘッド比$R_{\sigma}(p,n)$、及び並列効率$E_{para}(p,n)$を表示装置や印刷装置等に出力する（ステップＳ１３）。

【００３９】

これによりユーザは、並列効率を得るのと同時に、性能改善を阻害する要因の指標である並列化率、逐次計算時間比及び並列オーバーヘッド比の並列効率に及ぼす寄与を定量的に論じることができるようになる。

【００４０】

図１及び図２は式（７）により並列効率を計算する場合の機能ブロック及び処理フローを示していた。次に、図３及び図４を用いて式（９）により並列効率を

計算する場合の機能ブロック及び処理フローを説明する。なお、式（９）は、並列効率と、並列化率、逐次計算時間比及び並列オーバーヘッド比との関係が示されているわけではないが、経験的に精度良く並列効率を計算できることが分かっている。

【００４１】

図３は、並列計算機システムの並列効率を計算するための並列効率計算装置であるコンピュータ２の機能ブロック図を示している。コンピュータ２は、データ取得部２１と、前処理部２３と、指標データ計算部２４、並列効率計算部２５と、出力部２７と、表示装置や印刷装置等である出力装置２９とを含む。データ取得部２１は、図１に示したデータ取得部１１と同じ処理を実施する。前処理部２３は、並列処理部分の処理時間$\beta(p,n)$／ｐにプロセッサ数ｐを掛けて、並列処理部分を逐次処理した場合の処理時間$\beta(p,n)$を計算する。指標データ計算部２４は、データ取得部２１で取得された逐次処理部分の処理時間$\alpha(p,n)$と、並列処理のオーバーヘッドにより生ずる処理時間$\sigma(p,n)$と、並列処理時間$\tau(p,n)$と、前処理部２３により計算された$\beta(p,n)$とを用いて、並列化率$R_{para}(p,n)$と、逐次計算時間比$R_{\alpha}(p,n)$と、並列オーバーヘッド比$R_{\sigma}(p,n)$とを、式（４）、（５）及び（６）を用いて計算する。並列効率計算部２５は、データ取得部２１により取得された逐次処理部分の処理時間$\alpha(p,n)$、並列処理時間$\tau(p,n)$及びプロセッサ数ｐと、前処理部２３により計算された並列処理部分を逐次処理した場合の処理時間$\beta(p,n)$とを用いて、式（９）に従って並列効率$E_{para}(p,n)$を計算する。出力部２７は、並列効率計算部２５により計算された並列効率$E_{para}(p,n)$と、指標データ計算部２４により計算された並列化率$R_{para}(p,n)$、逐次計算時間比$R_{\alpha}(p,n)$及び並列オーバーヘッド比$R_{\sigma}(p,n)$とを表示装置や印刷装置等に出力する。

【００４２】

図４を用いて式（９）により並列効率$E_{para}(p,n)$を計算する場合の処理フローの一例を示す。データ取得部２１は、プロセッサ数ｐ、逐次処理部分の処理時間$\alpha(p,n)$、並列処理部分の処理時間$\beta(p,n)$／ｐ、並列処理のオーバーヘッドにより生ずる処理時間$\sigma(p,n)$、並列処理のオーバーヘッドにより生ずる処理時間

σ(p,n)及び並列処理時間τ(p,n)を取得し、メインメモリ等の記憶装置に格納する（ステップＳ２１）。また、前処理部３３は、並列処理部分の処理時間β(p,n)／ｐにプロセッサ数ｐを掛けて、並列処理部分を逐次処理した場合の処理時間β(p,n)を計算し、記憶装置に格納する（ステップＳ２３）。そして、指標データ計算部２４は、並列化率$R_{para}(p,n)$を式（４）に従って計算し、計算結果を記憶装置に格納する（ステップＳ２４）。また、指標データ計算部２４は、逐次計算時間比$R_{\alpha}(p,n)$を式（５）に従って計算し、計算結果を記憶装置に格納する（ステップＳ２５）。さらに、指標データ計算部２４は、並列オーバーヘッド比$R_{\sigma}(p,n)$を式（６）に従って計算し、計算結果を記憶装置に格納する（ステップＳ２６）。ステップＳ２４乃至ステップＳ２６の実行順番は任意である。また、以下で述べるステップＳ２７と並列に実行するような構成も可能である。

【００４３】

そして、並列効率計算部２５は、データ取得部２１により取得された逐次処理部分の処理時間α(p,n)と並列処理時間τ(p,n)とプロセッサ数ｐと、前処理部２３により計算された並列処理部分を逐次処理した場合の処理時間β(p,n)とを用いて、式（９）を用いて並列効率$E_{para}(p,n)$を計算し、記憶装置に格納する（ステップＳ２７）。そして、出力部２７は、ステップＳ２４乃至ステップＳ２７で計算した、並列化率$R_{para}(p,n)$、逐次計算時間比$R_{\alpha}(p,n)$、並列オーバーヘッド比$R_{\sigma}(p,n)$、及び並列効率$E_{para}(p,n)$を表示装置や印刷装置等に出力する（ステップＳ２８）。

【００４４】

このようにすれば、ユーザは並列計算機システムの並列効率を、一度の計測により短時間で簡単に得ることができるようになる。また、性能改善を阻害する要因の指標である並列化率、逐次計算時間比及び並列オーバーヘッド比の並列効率に及ぼす寄与を定量的に論じることができるようになる。さらに、並列効率の計算精度も式（７）及び（８）に比して同等か高くなる。

【００４５】

次に図５を用いて式（８）により並列効率を計算する場合の処理フローの一例を説明する。なお、この場合の並列効率計算装置であるコンピュータの機能ブロ

ック図は図１と同様である。但し、並列効率計算部１５は式（８）により並列効率を計算するようになっている。

【００４６】

データ取得部１１は、プロセッサ数ｐ、逐次処理部分の処理時間$\alpha(p,n)$、並列処理部分の処理時間$\beta(p,n)$／ｐ、並列処理のオーバーヘッドにより生ずる処理時間$\sigma(p,n)$及び並列処理時間$\tau(p,n)$を取得し、メインメモリ等の記憶装置に格納する（ステップＳ３１）。また、データ取得部１１は、並列処理部分の処理時間$\beta(p,n)$／ｐにプロセッサ数ｐを掛けて、並列処理部分を逐次処理した場合の処理時間$\beta(p,n)$を計算し、記憶装置に格納する（ステップＳ３３）。

【００４７】

次に、指標データ計算部１３は、並列化率$R_{para}(p,n)$を式（４）に従って計算し、計算結果を記憶装置に格納する（ステップＳ３５）。また、指標データ計算部１３は、逐次計算時間比$R_{\alpha}(p,n)$を式（５）に従って計算し、計算結果を記憶装置に格納する（ステップＳ３６）。さらに、指標データ計算部１３は、並列オーバーヘッド比$R_{\sigma}(p,n)$を式（６）に従って計算し、計算結果を記憶装置に格納する（ステップＳ３７）。ステップＳ３５乃至ステップＳ３７の実行順番は任意である。又、以下で述べるステップＳ３８と並列に実行するような構成も可能である。そして、並列効率計算部１５は、指標データ計算部１３により計算された並列化率$R_{para}(p,n)$と、データ取得部１１により取得された並列処理時間$\tau(p,n)$と並列処理部分の処理時間$\beta(p,n)$／ｐとを用いて式（８）に従って並列化率$E_{para}(p,n)$を計算し、記憶装置に格納する（ステップＳ３８）。出力部１７は、ステップＳ５乃至ステップＳ１１で計算した、並列化率$R_{para}(p,n)$、逐次計算時間比$R_{\alpha}(p,n)$、並列オーバーヘッド比$R_{\sigma}(p,n)$、及び並列効率$E_{para}(p,n)$を表示装置や印刷装置等に出力する（ステップＳ３９）。

【００４８】

これによりユーザは、並列計算機システムの並列効率を、一度の計測により短時間で簡単に得ることができるようになる。また、並列効率を得るのと同時に、性能改善を阻害する要因の指標である並列化率、逐次計算時間比及び並列オーバーヘッド比の並列効率に及ぼす寄与を定量的に論じることができるようになる。

【００４９】

以上本発明の一実施の形態を説明したが、本発明はこれに限定されるものではない。例えば、図１及び図３に示したブロック図は、必ずしもプログラムモジュールに対応するものではなく、別の分け方でモジュール化してもよい。また、処理フローについても同時に又は順番を入れ替えて実施することができる場合もある。

【００５０】

［具体例１］

時間測定により、逐次処理部分の処理時間$\alpha(p,n)$が８時間であり、並列処理部分の処理時間$\beta(p,n)$／ｐが１４時間であり、並列処理のためのオーバーヘッドにより生ずる処理時間$\sigma(p,n)$が１０時間であり、プロセッサ数が１００である場合に並列効率などを計算する。

$\beta(p,n)=\beta(p,n)/p\times p=14\times100=1400$

$R_{para}(p,n)\equiv\beta(p,n)/[\alpha(p,n)+\beta(p,n)]$

$=1400/[8+1400]=0.994$

$R_{\alpha}(p,n)\equiv\alpha(p,n)/\tau(p,n)$

$=8/(8+14+10)=0.25$

$R_{\sigma}(p,n)\equiv\sigma(p,n)/\tau(p,n)$

$=10/(8+14+10)=0.313$

【００５１】

これらの値を式（７）に代入すれば並列効率が求められる。

$E_{para}(p,n)=(1-0.25-0.313)/0.994$

$=0.440$

【００５２】

また式（８）を使用する場合には以下のようになる。

$E_{para}(p,n)=14/[0.994\cdot(8+14+10)]$

$=0.440$

【００５３】

さらに式（９）を使用する場合には以下のようになる。

$E_{para}(p,n)$＝（８＋１４００）／［（８＋１４＋１０）・１００］

＝０．４４０

【００５４】

［具体例２］

サンプリングの場合に、逐次処理のカウント回数が８回であり、並列処理のカウント回数が１４回であり、並列処理のためのオーバーヘッドにより生ずる処理のカウント回数が１０回であり、プロセッサ数が１００である場合に並列効率等を計算する。

$\beta(p,n)=\beta(p,n)/p\times p=14\times100=1400$

$R_{para}(p,n)\equiv\beta(p,n)/[\alpha(p,n)+\beta(p,n)]$

＝１４００／［８＋１４００］＝０．９９４

$R_{\alpha}(p,n)\equiv\alpha(p,n)/\tau(p,n)$

＝８／（８＋１４＋１０）＝０．２５

$R_{\sigma}(p,n)\equiv\sigma(p,n)/\tau(p,n)$

＝１０／（８＋１４＋１０）＝０．３１３

【００５５】

これらの値を式（７）に代入すれば並列効率が求められる。

$E_{para}(p,n)$＝（１－０．２５－０．３１３）／０．９９４

＝０．４４０

【００５６】

また式（８）を使用する場合には以下のようになる。

$E_{para}(p,n)$＝１４／［０．９９４・（８＋１４＋１０）］

＝０．４４０

【００５７】

さらに式（９）を使用する場合には以下のようになる。

$E_{para}(p,n)$＝（８＋１４００）／［（８＋１４＋１０）・１００］

＝０．４４０

このように時間測定の場合と同じ結果になる。

【００５８】

［具体例３］

ｐ＝１０で、逐次処理部分の処理時間α(p,n)が１００分、並列処理部分の処理時間β(p,n)／ｐが１分、並列処理のためのオーバーヘッドにより生じた処理時間σ(p,n)が１分、並列処理時間τ(p,n)が１０２分と時間測定された場合、β(p,n)＝１×１０＝１０となる。また、式（４）、式（５）及び式（６）に代入することにより、以下の値を得る。

$R_{para}(p,n)$＝１０／（１００＋１０）＝０．０９１

$R_{\alpha}(p,n)$＝１００／１０２＝０．９８０

$R_{\sigma}(p,n)$＝１／１０２＝０．００９８

【００５９】

例えば式（９）で並列効率を計算すると、以下のようになる。

$E_{para}(p,n)$＝（１００＋１０）／１２０／１０＝０．１０８

【００６０】

これらの値から並列処理の性能評価を行うと、並列処理を阻害する要因は$R_{\alpha}(p,n)$が大きいためである。$R_{para}(p,n)$が小さいことから、プロセッサ数が増して相対的に$R_{\alpha}(p,n)$が大きく見えるのではなく、たとえｐ＝２としても並列性能がでないことがわかる。

【００６１】

［具体例４］

ｐ＝１０で、逐次処理部分の処理時間α(p,n)が１分、並列処理部分の処理時間β(p,n)／ｐが１００分、並列処理のためのオーバーヘッドにより生じた処理時間σ(p,n)が１分、並列処理時間τ(p,n)が１０２分と時間測定された場合、β(p,n)＝１００×１０＝１０００となる。また、式（４）、式（５）及び式（６）に代入することにより、以下の値を得る。

$R_{para}(p,n)$＝１０００／（１０＋１０００）＝０．９９９

$R_{\alpha}(p,n)$＝１／１０２＝０．００９８

$R_{\sigma}(p,n)$＝１／１０２＝０．００９８

【００６２】

例えば式（８）で並列効率を計算すると、以下のようになる。

$E_{para}(p,n)$＝１／０．９９９×１／１０２×１００＝０．９８１

【００６３】

これらの値から並列処理の性能評価を行うと、並列効率０．９８１と非常に高い状態で並列処理を実施していることが分かる。

【００６４】

[具体例５]

ｐ＝１０で、逐次処理部分の処理時間$\alpha(p,n)$が１０分、並列処理部分の処理時間$\beta(p,n)$／ｐが１０分、並列処理のためのオーバーヘッドにより生じた処理時間$\sigma(p,n)$が１０分、並列処理時間$\tau(p,n)$が３０分と時間測定された場合、$\beta(p,n)$＝１０×１０＝１００となる。また、式（４）、式（５）及び式（６）に代入することにより、以下の値を得る。

$R_{para}(p,n)$＝１００／（１０＋１００）＝０．９０９

$R_{\alpha}(p,n)$＝１０／３０＝０．３３３

$R_{\sigma}(p,n)$＝１０／３０＝０．３３３

【００６５】

例えば式（７）で並列効率を計算すると、以下のようになる。

$E_{para}(p,n)$＝（１－0.333－0.333）／０．９０９＝０．３６７

【００６６】

これらの値から並列処理の性能評価を行うと、並列処理を阻害する要因は$R_{\alpha}(p,n)$及び$R_{\sigma}(p,n)$であり、０．３３３であり、同じ割合で阻害している。

【００６７】

[具体例６]

ｐ＝１０で、逐次処理のカウント数が１００００で、並列処理のカウント数が１００で、並列処理のためのオーバーヘッドにより生ずる処理のカウント数が１００であり、全体のカウント数が１０２００とサンプリングされた場合、$\beta(p,n)$＝１００×１０＝１０００となる。また、式（４）、式（５）及び式（６）に代入することにより、以下の値を得る。

$R_{para}(p,n)$＝１０００／（１００００＋１０００）＝０．０９１

$R_{\alpha}(p,n)$＝１００００／１０２００＝０．９８０

$R_{\sigma}(p,n)$＝１００／１０２００＝０．００９８

【００６８】

例えば式（９）で並列効率を計算すると、以下のようになる。

$E_{para}(p,n)$＝（10000＋1000）／10200／10＝０．１０８

【００６９】

これらの値から並列処理の性能評価を行うと、並列処理を阻害する要因は$R_{\alpha}(p,n)$が大きいためである。$R_{para}(p,n)$が小さいことから、プロセッサ数が増加して相対的に$R_{\alpha}(p,n)$が大きく見えるのではなく、例えｐ＝２としても並列性能がでないことが分かる。

【００７０】

［具体例７］

ｐ＝１０で、逐次処理のカウント数が１００で、並列処理のカウント数が１００００で、並列処理のためのオーバーヘッドにより生ずる処理のカウント数が１００であり、全体のカウント数が１０２００とサンプリングされた場合、$\beta(p,n)$＝１００００×１０＝１０００００となる。また、式（４）、式（５）及び式（６）に代入することにより、以下の値を得る。

$R_{para}(p,n)$＝100000／（100＋100000）＝０．９９９

$R_{\alpha}(p,n)$＝１００／１０２００＝０．００９８

$R_{\sigma}(p,n)$＝１００／１０２００＝０．００９８

【００７１】

例えば式（８）で並列効率を計算すると、以下のようになる。

$E_{para}(p,n)$＝１／0.999・１／10200・10000＝０．９８１

【００７２】

これらの値から並列処理の性能評価を行うと、並列計算機システムは並列効率０．９８１と高い状態で並列処理を実行していることが分かる。

【００７３】

［具体例８］

ｐ＝１０で、逐次処理のカウント数が１０００で、並列処理のカウント数が１０００で、並列処理のためのオーバーヘッドにより生ずる処理のカウント数が１

０００であり、全体のカウント数が３０００とサンプリングされた場合、$\beta(p,n)$＝１０００×１０＝１００００となる。また、式（４）、式（５）及び式（６）に代入することにより、以下の値を得る。

$R_{para}(p,n)$＝10000／（1000＋10000）＝０．９０９

$R_{\alpha}(p,n)$＝１０００／３０００＝０．３３３

$R_{\sigma}(p,n)$＝１０００／３０００＝０．３３３

【００７４】

例えば式（７）で並列効率を計算すると、以下のようになる。

$E_{para}(p,n)$＝（１－0.333－0.333）／0.909＝０．３６７

【００７５】

これらの値から並列処理の性能評価を行うと、並列処理を阻害する要因は$R_{\alpha}(p,n)$及び$R_{\sigma}(p,n)$であり、０．３３３であり、同じ割合で阻害していることが分かる。

【００７６】

（付記１）

並列計算機システムの並列効率を計算する方法であって、

並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、前記並列処理プログラム実行時における並列処理部分の処理時間に関する情報と、並列処理のオーバーヘッドにより生ずる処理時間に関する情報とを取得し、記憶装置に格納するステップと、

取得された前記逐次処理部分の処理時間に関する情報と前記並列処理部分の処理時間に関する情報と前記並列処理のオーバーヘッドにより生ずる処理時間に関する情報とを用いて、並列化率と、逐次計算時間比と、並列オーバーヘッド比とを計算し、記憶装置に格納する指標計算ステップと、

計算された前記並列化率と前記逐次計算時間比と前記並列オーバーヘッド比とを用いて並列効率を計算し、記憶装置に格納する並列効率計算ステップと、

を含む並列効率計算方法。

【００７７】

（付記２）

前記逐次処理部分の処理時間に関する情報が、前記並列処理プログラム実行期間内の所定周期毎の実行状況確認において逐次処理が実施されていると判断された回数であり、＊　前記並列処理部分の処理時間に関する情報が、前記並列処理プログラム実行期間内の所定周期毎の実行状況確認において並列処理が実施されていると判断された回数であり、

前記並列処理のオーバーヘッドにより生ずる処理時間に関する情報が、前記並列処理プログラム実行期間内の所定周期毎の実行状況確認において前記並列処理のオーバーヘッドにより生ずる処理が実施されていると判断された回数である

ことを特徴とする付記１記載の並列効率計算方法。

【００７８】

（付記３）

前記指標計算ステップが、

前記並列処理部分の処理時間に関する情報をプロセッサ数倍し、前記並列処理プログラム実行時における並列処理部分の逐次処理時における処理時間に関する情報として記憶装置に格納するステップと、

（前記並列処理部分の逐次処理時における処理時間に関する情報）／（前記逐次処理部分の処理時間に関する情報＋前記並列処理部分の逐次処理時における処理時間に関する情報）を計算し、前記並列化率として記憶装置に格納するステップと、

を含む付記１記載の並列効率計算方法。

【００７９】

（付記４）

前記指標計算ステップが、

（前記逐次処理部分の処理時間に関する情報）／（前記並列処理プログラムの全処理時間に関する情報）を計算し、前記逐次計算時間比として記憶装置に格納するステップ、

を含む付記１記載の並列効率計算方法。

【００８０】

（付記５）

前記指標計算ステップが、

（前記並列処理のオーバーヘッドにより生ずる処理時間に関する情報）／（前記並列処理プログラムの全処理時間に関する情報）を計算し、前記並列オーバーヘッド比として記憶装置に格納するステップ、

を含む付記１記載の並列効率計算方法。

【００８１】

（付記６）

前記並列効率計算ステップが、

１／（前記並列化率）×（１－（前記逐次計算時間比）－（前記並列オーバーヘッド比））を計算し、前記並列効率として記憶装置に格納するステップ、

であることを特徴とする付記１記載の並列効率計算方法。

【００８２】

（付記７）

並列計算機システムの並列効率を計算する方法であって、

並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、前記並列処理プログラム実行時における並列処理部分の処理時間に関する情報と前記並列処理プログラムの全処理時間に関する情報とを取得し、記憶装置に格納するステップと、

取得された前記並列処理部分の処理時間に関する情報をプロセッサ数倍し、前記並列処理プログラム実行時における並列処理部分の逐次処理時における処理時間に関する情報として記憶装置に格納するステップと、

取得された前記逐次処理部分の処理時間に関する情報と前記並列処理部分の処理時間に関する情報とを少なくとも用いて、並列化率と、逐次計算時間比と、並列オーバーヘッド比とを計算し、記憶装置に格納する指標計算ステップと、

（（前記逐次処理部分の処理時間に関する情報）＋（前記並列処理部分の逐次処理時における処理時間に関する情報））／（（前記並列処理プログラムの全処理時間に関する情報）×（前記プロセッサ数））を計算し、並列効率として記憶装置に格納するステップと、

を含む並列効率計算方法。

【００８３】

（付記８）

並列計算機システムの並列効率を計算する方法であって、

並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、前記並列処理プログラム実行時における並列処理部分の処理時間に関する情報と前記並列処理プログラムの全処理時間に関する情報とを取得し、記憶装置に格納するステップと、

取得された前記逐次処理部分の処理時間に関する情報と前記並列処理部分の処理時間に関する情報とを用いて並列化率を計算し、記憶装置に格納するステップと、

前記並列化率の逆数と前記並列処理プログラムの全処理時間に関する情報の逆数と前記並列処理部分の処理時間に関する情報との積を計算し、並列効率として記憶装置に格納するステップと、

を含む並列効率計算方法。

【００８４】

（付記９）

並列計算機システムの並列効率を計算するプログラムであって、

並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、前記並列処理プログラム実行時における並列処理部分の処理時間に関する情報と、並列処理のオーバーヘッドにより生ずる処理時間に関する情報とを取得し、記憶装置に格納するステップと、

取得された前記逐次処理部分の処理時間に関する情報と前記並列処理部分の処理時間に関する情報と前記並列処理のオーバーヘッドにより生ずる処理時間に関する情報とを用いて、並列化率と、逐次計算時間比と、並列オーバーヘッド比とを計算し、記憶装置に格納する指標計算ステップと、

計算された前記並列化率と前記逐次計算時間比と前記並列オーバーヘッド比とを用いて並列効率を計算し、記憶装置に格納する並列効率計算ステップと、

をコンピュータに実行させるためのプログラム。

【００８５】

（付記１０）

並列計算機システムの並列効率を計算するプログラムであって、

並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、前記並列処理プログラム実行時における並列処理部分の処理時間に関する情報と前記並列処理プログラムの全処理時間に関する情報とを取得し、記憶装置に格納するステップと、

取得された前記並列処理部分の処理時間に関する情報をプロセッサ数倍し、前記並列処理プログラム実行時における並列処理部分の逐次処理時における処理時間に関する情報として記憶装置に格納するステップと、

取得された前記逐次処理部分の処理時間に関する情報と前記並列処理部分の処理時間に関する情報とを少なくとも用いて、並列化率と、逐次計算時間比と、並列オーバーヘッド比とを計算し、記憶装置に格納する指標計算ステップと、

（（前記逐次処理部分の処理時間に関する情報）＋（前記並列処理部分の逐次処理時における処理時間に関する情報））／（（前記並列処理プログラムの全処理時間に関する情報）×（前記プロセッサ数））を計算し、並列効率として記憶装置に格納するステップと、

をコンピュータに実行させるためのプログラム。

【００８６】

（付記１１）

並列計算機システムの並列効率を計算するプログラムであって、

並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、前記並列処理プログラム実行時における並列処理部分の処理時間に関する情報と前記並列処理プログラムの全処理時間に関する情報とを取得し、記憶装置に格納するステップと、

取得された前記逐次処理部分の処理時間に関する情報と前記並列処理部分の処理時間に関する情報とを用いて並列化率を計算し、記憶装置に格納するステップと、

前記並列化率の逆数と前記並列処理プログラムの全処理時間に関する情報の逆数と前記並列処理部分の処理時間に関する情報との積を計算し、並列効率として

記憶装置に格納するステップと、

をコンピュータに実行させるためのプログラム。

【００８７】

（付記１２）

付記９乃至１１のいずれか１つ記載のプログラムを格納した記録媒体。

【００８８】

（付記１３）

並列計算機システムの並列効率を計算する装置であって、

並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、前記並列処理プログラム実行時における並列処理部分の処理時間に関する情報と、並列処理のオーバーヘッドにより生ずる処理時間に関する情報とを取得し、記憶装置に格納する手段と、

取得された前記逐次処理部分の処理時間に関する情報と前記並列処理部分の処理時間に関する情報と前記並列処理のオーバーヘッドにより生ずる処理時間に関する情報とを用いて、並列化率と、逐次計算時間比と、並列オーバーヘッド比とを計算し、記憶装置に格納する指標計算手段と、

計算された前記並列化率と前記逐次計算時間比と前記並列オーバーヘッド比とを用いて並列効率を計算し、記憶装置に格納する並列効率計算手段と、

を有する並列効率計算装置。

【００８９】

（付記１４）

並列計算機システムの並列効率を計算する装置であって、

並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、前記並列処理プログラム実行時における並列処理部分の処理時間に関する情報と前記並列処理プログラムの全処理時間に関する情報とを取得し、記憶装置に格納する手段と、

取得された前記並列処理部分の処理時間に関する情報をプロセッサ数倍し、前記並列処理プログラム実行時における並列処理部分の逐次処理時における処理時間に関する情報として記憶装置に格納する手段と、

取得された前記逐次処理部分の処理時間に関する情報と前記並列処理部分の処理時間に関する情報とを少なくとも用いて、並列化率と、逐次計算時間比と、並列オーバーヘッド比とを計算し、記憶装置に格納する指標計算手段と、

（（前記逐次処理部分の処理時間に関する情報）＋（前記並列処理部分の逐次処理時における処理時間に関する情報））／（（前記並列処理プログラムの全処理時間に関する情報）×（前記プロセッサ数））を計算し、並列効率として記憶装置に格納する手段と、

を有する並列効率計算装置。

【００９０】

（付記１５）

並列計算機システムの並列効率を計算する装置であって、

並列処理プログラム実行時における逐次処理部分の処理時間に関する情報と、前記並列処理プログラム実行時における並列処理部分の処理時間に関する情報と前記並列処理プログラムの全処理時間に関する情報とを取得し、記憶装置に格納する手段と、

取得された前記逐次処理部分の処理時間に関する情報と前記並列処理部分の処理時間に関する情報とを用いて並列化率を計算し、記憶装置に格納する手段と、

前記並列化率の逆数と前記並列処理プログラムの全処理時間に関する情報の逆数と前記並列処理部分の処理時間に関する情報との積を計算し、並列効率として記憶装置に格納する手段と、

を含む並列効率計算方法。

【００９１】

【発明の効果】

以上のように本発明のよれば、並列効率の値と性能改善を阻害する要因を定量的に結び付けることにより性能を阻害している原因を明確にするための技術を提供することができる。

【００９２】

また、並列効率を１回の測定結果により計算できるようにするための技術を提供することができる。

【００９３】

さらに、より精度よく並列効率を計算するできるようにするための技術を提供することができる。

【００９４】

また、並列計算機システムの性能評価を容易にし且つ評価に要する時間を短縮できるようにするための技術を提供することもできる。

【図面の簡単な説明】

【図１】

本発明の一実施の形態に係るブロック図である。

【図２】

並列効率を計算するための第１の処理フローを示す図である。

【図３】

本発明の一実施の形態に係るブロック図である。

【図４】

並列効率を計算するための第２の処理フローを示す図である。

【図５】

並列効率を計算するための第３の処理フローを示す図である。

【符号の説明】

１，２　コンピュータ

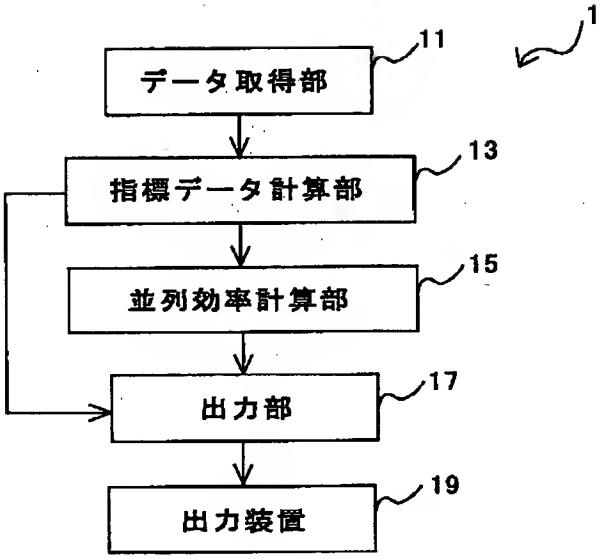
１１，２１　データ取得部　　１３　指標データ計算部

１５，２５　並列効率計算部　１７，２７　出力部

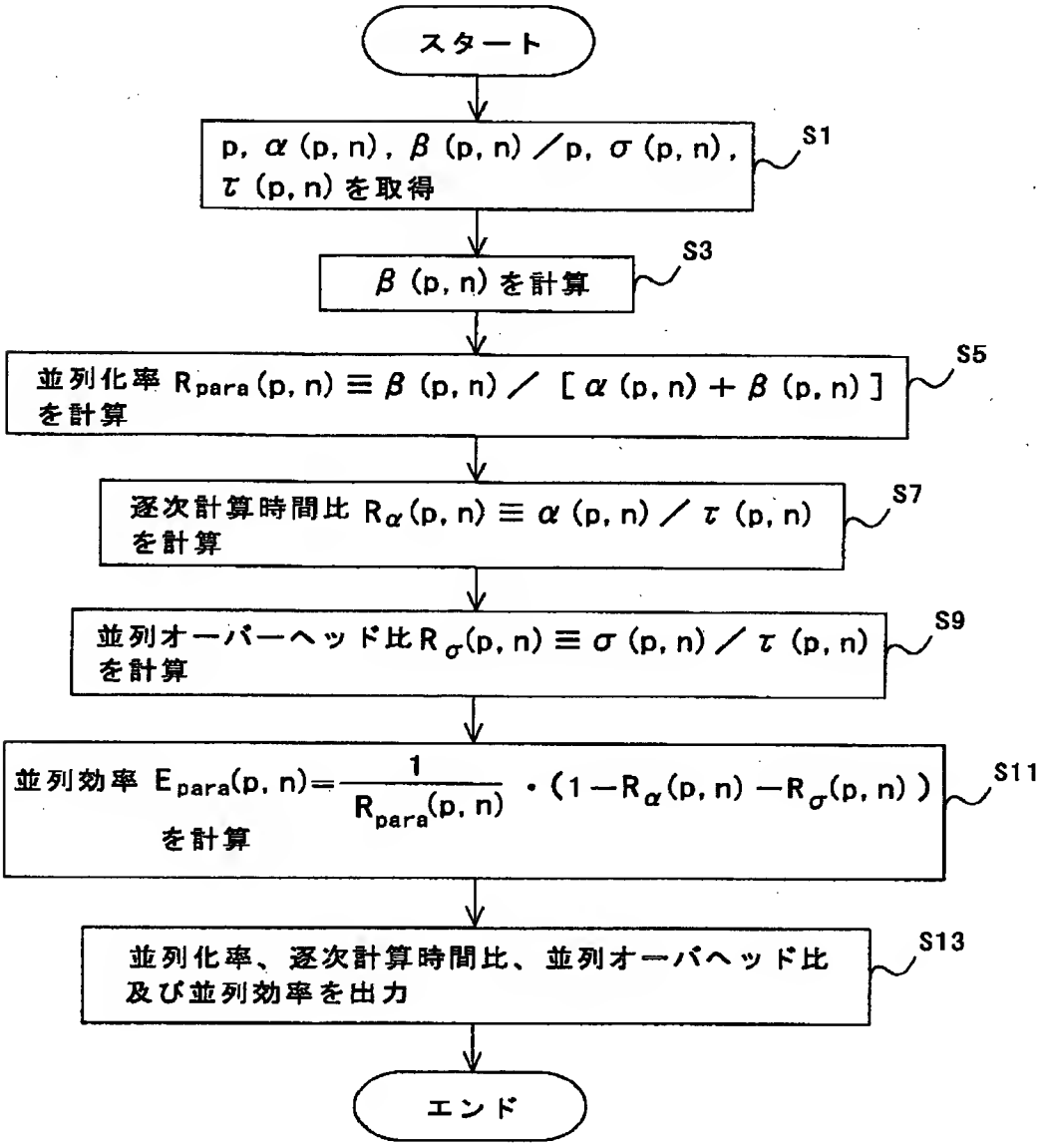
１９，２９　出力装置　　２３　前処理部

【書類名】　　図面

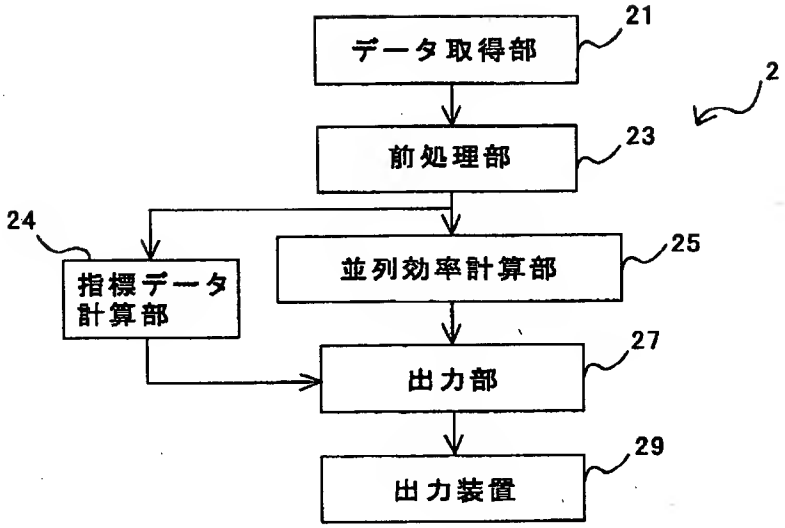
【図１】



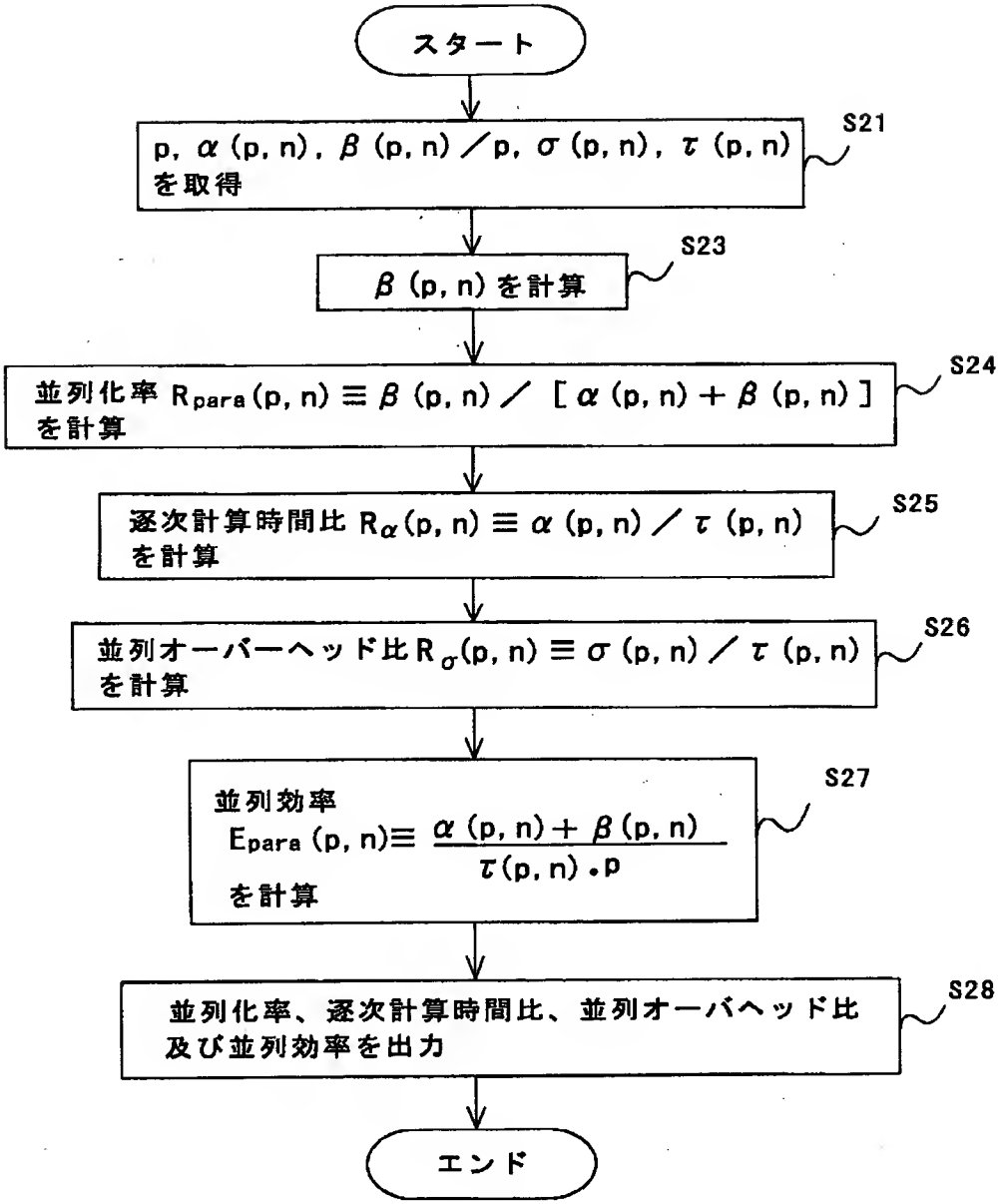
【図２】


スタート
p, α(p, n), β(p, n) ／ p, σ(p, n), τ(p, n) を取得
S1
β(p, n) を計算
S3
並列化率 $R_{para}(p, n) \equiv \beta(p, n) / [\alpha(p, n) + \beta(p, n)]$ を計算
S5
逐次計算時間比 $R_{\alpha}(p, n) \equiv \alpha(p, n) / \tau(p, n)$ を計算
S7
並列オーバーヘッド比 $R_{\sigma}(p, n) \equiv \sigma(p, n) / \tau(p, n)$ を計算
S9
並列効率 $E_{para}(p, n) = \frac{1}{R_{para}(p, n)} \cdot (1 - R_{\alpha}(p, n) - R_{\sigma}(p, n))$ を計算
S11
並列化率、逐次計算時間比、並列オーバヘッド比及び並列効率を出力
S13
エンド

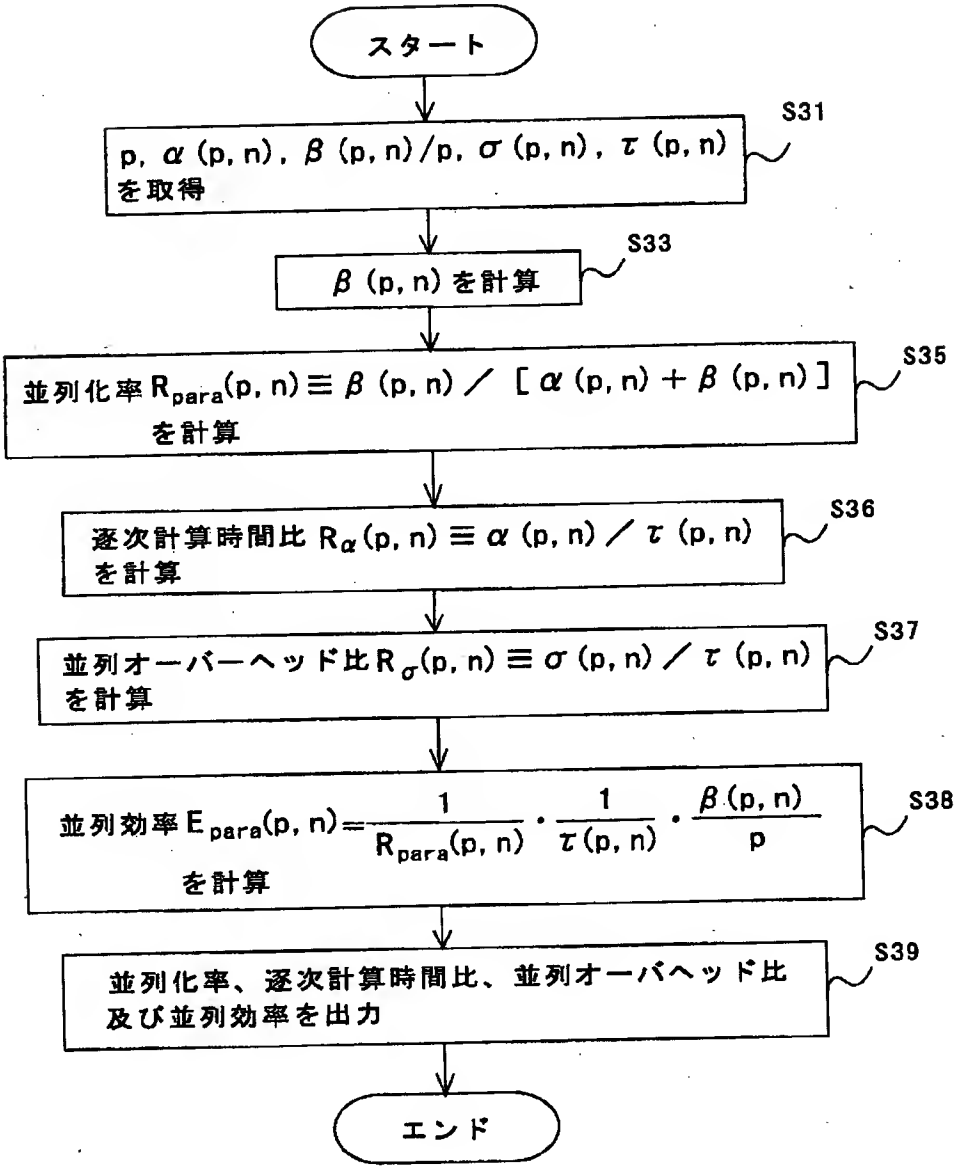
【図３】



【図４】


スタート
p, α(p, n), β(p, n) ／ p, σ(p, n), τ(p, n) を取得
S21
β(p, n) を計算
S23
並列化率 Rpara(p, n) ≡ β(p, n) ／ [α(p, n) + β(p, n)] を計算
S24
逐次計算時間比 Rα(p, n) ≡ α(p, n) ／ τ(p, n) を計算
S25
並列オーバーヘッド比 Rσ(p, n) ≡ σ(p, n) ／ τ(p, n) を計算
S26
並列効率 Epara(p, n) ≡ (α(p, n) + β(p, n)) / (τ(p, n)・p) を計算
S27
並列化率、逐次計算時間比、並列オーバヘッド比及び並列効率を出力
S28
エンド

【図５】


スタート
S31
p, α(p, n), β(p, n)/p, σ(p, n), τ(p, n) を取得
S33
β(p, n) を計算
S35
並列化率 $R_{para}(p, n) \equiv \beta(p, n) / [\alpha(p, n) + \beta(p, n)]$ を計算
S36
逐次計算時間比 $R_{\alpha}(p, n) \equiv \alpha(p, n) / \tau(p, n)$ を計算
S37
並列オーバーヘッド比 $R_{\sigma}(p, n) \equiv \sigma(p, n) / \tau(p, n)$ を計算
S38
並列効率 $E_{para}(p, n) = \frac{1}{R_{para}(p, n)} \cdot \frac{1}{\tau(p, n)} \cdot \frac{\beta(p, n)}{p}$ を計算
S39
並列化率、逐次計算時間比、並列オーバヘッド比及び並列効率を出力
エンド

【書類名】　　要約書

【要約】

【課題】

並列効率の値と性能改善を阻害する要因を定量的に結び付けることにより性能を阻害している原因を明確にする。

【解決手段】

並列処理プログラム実行時における逐次処理部分の処理時間$\alpha(p,n)$と、並列処理部分の処理時間$\beta(p,n)/p$と、並列処理のオーバーヘッドにより生ずる処理時間$\sigma(p,n)$とを取得し、取得された逐次処理部分の処理時間$\alpha(p,n)$と並列処理部分の処理時間$\beta(p,n)/p$と並列処理のオーバーヘッドにより生ずる処理時間$\sigma(p,n)$とを用いて、並列化率$R_{para}(p,n)$と、逐次計算時間比$R_{\alpha}(p,n)$と、並列オーバーヘッド比$R_{\sigma}(p,n)$とを計算する。そして、$1/R_{para}(p,n)\times(1-R_{\alpha}(p,n)-R_{\sigma}(p,n))$により並列効率$E_{para}(p,n)$を計算する。

出証特２００１－３０８７６５５

出　願　人　履　歴　情　報

識別番号　　〔０００００５２２３〕

１．変更年月日　　１９９６年　３月２６日
［変更理由］　住所変更
住　所　神奈川県川崎市中原区上小田中４丁目１番１号
氏　名　富士通株式会社